(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－116838
（P2002－116838A）
(43)公開日　平成14年４月19日(2002.4.19)

(51)Int.Cl.⁷		識別記号	FI		テーマコード（参考）
Ｇ０６Ｆ	1/00		Ｇ０６Ｆ 12/00	５１０Ｂ	５Ｂ０１７
	9/445			５３７Ｈ	５Ｂ０７６
	12/00	５１０	12/14	３２０Ｂ	５Ｂ０８２
		５３７	13/00	５３０Ｂ	５Ｂ０８５
	12/14	３２０	15/00	３３０Ｚ	５Ｊ１０４

審査請求　有　請求項の数19　ＯＬ　（全 11 頁）　最終頁に続く

(21)出願番号　特願2001－195753(P2001－195753)

(22)出願日　平成13年６月28日(2001.6.28)

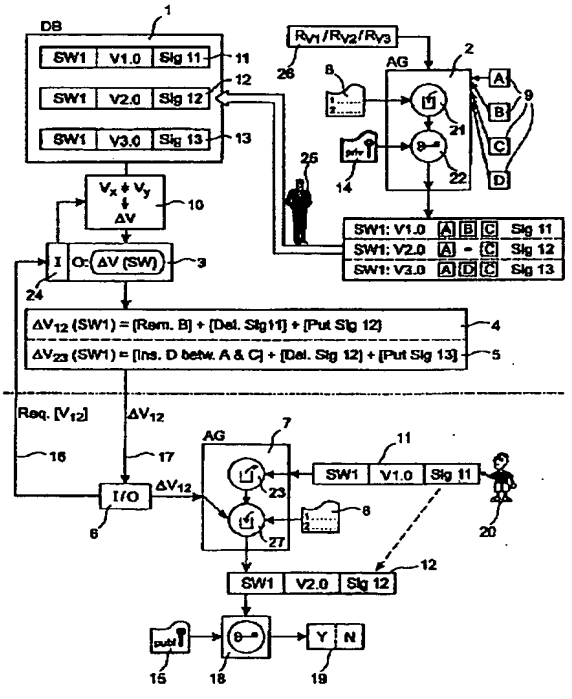
(31)優先権主張番号　００１１３８５７．７
(32)優先日　平成12年６月30日(2000.6.30)
(33)優先権主張国　欧州特許庁（ＥＰ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレーション
ＩＮＴＥＲＮＡＴＩＯＮＡＬ　ＢＵＳＩＮＥＳＳ　ＭＡＳＣＨＩＮＥＳ　ＣＯＲＰＯＲＡＴＩＯＮ
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）
(74)代理人　100086243
弁理士　坂口　博　（外２名）

最終頁に続く

(54)【発明の名称】　コードを更新するためのデバイスおよび方法

(57)【要約】　（修正有）
【課題】　伝送するデータの量を低減するが、機能性における変更を容易にし、Ｊａｖａコード・コンテナなど、署名付きコードのセキュリティ特性を保持する方法を提供する。
【解決手段】　ソフトウェア取得エンティティ２０が既存の第１の署名付きコード片１１から第２の署名付きコード片１２に達することができるようにする、ソフトウェア・プロバイダのための方法を対象とする。両方のコード片は、ソフトウェア・プロバイダ側で、第１のソフトウェア・アーカイブ生成器２を使用することによって、生成命令の使用下で生成された。ソフトウェア・プロバイダは、第１の署名付きコード片１１から第２の署名付きコード片１２に達するために必要なステップを含む差分コード４をソフトウェア取得エンティティ２０に提供する。



1

【特許請求の範囲】

【請求項１】ソフトウェア取得エンティティ（２０）が既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達することができるようにする、ソフトウェア・プロバイダ（２５）のための方法であって、両方のコード片（１１、１２、１３）は、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、前記第１の署名付きコード片（１１）から前記第２の署名付きコード片（１２、１３）に達するために必要なステップを含む差分コード（４、５）を前記ソフトウェア取得エンティティ（２０）に提供するステップを含み、この差分コード（４、５）は、前記ソフトウェア取得エンティティ（２０）側で、第２のソフトウェア・アーカイブ生成器（７）によって前記第１の署名付きコード片（１１）と結合されて前記第２の署名付きコード片（１２、１３）を生成するように使用可能であり、それにより、前記第１のソフトウェア・アーカイブ生成器（２）によって両方のコード片（１１、１２、１３）の前記生成のために使用されたそれらの生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給される方法。

【請求項２】前記生成命令（８）が、前記ソフトウェア・プロバイダ（２５）によって、好ましくは前記第２のソフトウェア・アーカイブ生成器（７）と共に前記ソフトウェア取得エンティティ（２０）に提供される、請求項１に記載の方法。

【請求項３】前記コード片（１１、１２、１３）が、秘密鍵（１４）を使用して署名される、請求項１または２に記載の方法。

【請求項４】前記署名付きコード片（１１、１２、１３）が、前記ソフトウェア・プロバイダ（２５）側の記憶装置（１）に格納される、請求項１ないし３のいずれか一項に記載の方法。

【請求項５】前記差分コード（４、５）が、好ましくは前記第１のソフトウェア・アーカイブ生成器（２）によって作成され、前記第１のソフトウェア・アーカイブ生成器（２）が前記第２の署名付きコード片（１２、１３）を生成する、請求項１ないし４のいずれか一項に記載の方法。

【請求項６】３つ以上のコード片（１１、１２、１３）が格納される場合、前記差分コード（４、５）が前記コード片（１１、１２、１３）の部分集合の間でのみ生成される、請求項１ないし５のいずれか一項に記載の方法。

【請求項７】前記第１のコード片（１１）から前記第２のコード片（１３）に達するために、いくつかの差分コード（４、５）が必要とされ、これらの差分コード（４、５）が単一の差分コードに併合されて前記ソフトウェア取得エンティティ（２０）に提供される、請求項６に記載の方法。

2

【請求項８】前記第１および第２のコード片（１１、１２、１３）が、前記ソフトウェア・プロバイダ（２５）側で、前記ソフトウェア取得エンティティ（２０）から受信された要求（１６）から対応する識別子を導出することによって識別される、請求項１ないし７のいずれか一項に記載の方法。

【請求項９】既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達するソフトウェア取得エンティティ（２０）のための方法であって、両方のコード片（１１、１２、１３）は、ソフトウェア・プロバイダ（２５）側で、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、

前記第１の署名付きコード片（１１）から前記第２の署名付きコード片（１２、１３）に達するために必要なステップを含む差分コード（４、５）の送達のために、コード修正要求（１６）を前記ソフトウェア・プロバイダ（２５）に送信するステップと、

前記差分コード（４、５）を受信するステップと、

第２のソフトウェア・アーカイブ生成器（７）を使用することによって前記差分コード（４、５）を前記第１の署名付きコード片（１１）と結合し、それにより前記第２の署名付きコード片（１２、１３）を生成するステップとを含み、それにより、前記第１のソフトウェア・アーカイブ生成器（２）によって両方のコード片（１１、１２、１３）の前記生成のために使用されたそれらの生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給される方法。

【請求項１０】前記生成命令（８）が前記ソフトウェア・プロバイダ（２５）から、好ましくは前記第２のソフトウェア・アーカイブ生成器（７）と共に受信される、請求項９に記載の方法。

【請求項１１】前記コード片（１１、１２、１３）が、秘密鍵（１４）を使用することによって署名され、署名（Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３）が、対応する公開鍵（１５）を使用することによって検証可能である、請求項９または１０に記載の方法。

【請求項１２】前記第１および第２のコード片（１１、１２、１３）が、前記コード修正要求（１６）において対応する識別子を与えることによって前記ソフトウェア取得エンティティ（２０）によって識別される、請求項９ないし１１のいずれか一項に記載の方法。

【請求項１３】請求項１ないし１２のいずれか一項に記載の方法を実行するためのプログラム・コード手段を含む、コンピュータ・プログラム製品。

【請求項１４】コンピュータ可読媒体上に格納された前記プログラム・コード手段を含む、請求項１３に記載のコンピュータ・プログラム製品。

【請求項１５】ソフトウェア取得エンティティ（２０）

が、既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達することができるようにするコード修正イネーブラであって、両方のコード片（１１、１２、１３）は、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、

前記第１の署名付きコード片（１１）から前記第２の署名付きコード片（１２、１３）に達するために必要なステップを含む差分コード（４、５）を生成するための差分コード生成器（１０）を含み、この差分コード（４、５）は、前記ソフトウェア取得エンティティ（２０）側で、第２のソフトウェア・アーカイブ生成器（７）によって前記第１の署名付きコード片（１１）と結合されて前記第２の署名付きコード片（１２、１３）を生成するように使用可能であり、それにより、前記生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給され、さらに、

前記ソフトウェア取得エンティティ（２０）に前記差分コード（４、５）を提供するための出力ユニット（３）を含む、コード修正イネーブラ。

【請求項１６】前記差分コード（４、５）の送達のためのコード修正要求（１６）を前記ソフトウェア取得エンティティ（２０）から受信するための入力ユニット（２４）をさらに含む、請求項１５に記載のコード修正イネーブラ。

【請求項１７】前記コード片（１１、１２、１３）を生成命令（８）の使用下で生成するための第１のソフトウェア・アーカイブ生成器（２）をさらに含む、請求項１５または１６に記載のコード修正イネーブラ。

【請求項１８】既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達するためのコード修正デバイスであって、両方のコード片（１１、１２、１３）は、ソフトウェア・プロバイダ（２５）側で、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、

受信された差分コード（４、５）を前記第１の署名付きコード片（１１）と結合し、それにより前記第２の署名付きコード片（１２、１３）を生成するための第２のソフトウェア・アーカイブ生成器（７）を含み、それにより、前記第１のソフトウェア・アーカイブ生成器（２）によって両方のコード片（１１、１２、１３）の前記生成のために使用されたそれらの生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給される、コード修正デバイス。

【請求項１９】コード修正要求（１６）を前記ソフトウェア・プロバイダ（２５）に送信し、前記差分コード（４、５）を受信するための入出力ユニット（６）をさらに含む、請求項１８に記載のコード修正デバイス。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、ソフトウェア取得エンティティが既存の第１の署名付きコード片から第２の署名付きコード片に達することができるようにする、ソフトウェア・プロバイダのための方法に関する。本発明は、既存の第１の署名付きコード片から第２の署名付きコード片に達するソフトウェア取得エンティティのための方法にも関する。

【０００２】

【従来の技術】今日の典型的なＪａｖａアプリケーションは、非常に大きく、数千個ものクラスからなる傾向がある。さらに、Ｊａｖａコードを実行する、普及しているウェブブラウザでは、異なるＪａｖａパッケージングおよび署名フォーマットが必要なので、コードが通常２度配布され、それによりさらにアプリケーションのサイズが増大する。このように複雑さが増すことは、より多くのプログラミングのエラーが、プログラムがすでに顧客へロール・アウトされたときになって初めて明らかになることに通じる。これにより、誤り訂正を顧客に送ることが必要となる。Ｊａｖａプログラムは、暗号化署名付きコンテナにおいて出荷されるので、これまで、完全なＪａｖａコンテナは、場合によっては数千個ものクラスを含み、その中のただ１つのクラス・ファイルが変更されたときでさえ再配布する必要がある。

【０００３】

【発明が解決しようとする課題】独立クレームによる本発明の目的は、伝送するデータの量を低減するが、機能性における変更を容易にし、Ｊａｖａコード・コンテナなど、署名付きコードのセキュリティ特性を保持する方法を提供することである。

【０００４】

【課題を解決するための手段】したがって、本明細書で解決される主な問題は、たとえば、コードにおけるエラーを修正する、あるいはコードを新しい機能セットへアップグレードするなど、第１のコード片を修正して第２のコード片に達するときに、顧客に出荷されるデータの量の低減である。

【０００５】よって、本発明は、ソフトウェア取得エンティティが既存の第１の署名付きコード片から第２の署名付きコード片に達することができるようにする、ソフトウェア・プロバイダのための方法を対象とする。両方のコード片は、ソフトウェア・プロバイダ側で、第１のソフトウェア・アーカイブ生成器を使用することによって、生成命令の使用下で生成された。ソフトウェア・プロバイダは、第１の署名付きコード片から第２の署名付きコード片に達するために必要なステップを含む差分コードをソフトウェア取得エンティティに提供する。差分コードは、ソフトウェア取得エンティティ側で、第２のソフトウェア・アーカイブ生成器によって第１の署名付きコード片と結合可能であり、第２の署名付きコード片

を生成する。したがって、第１のソフトウェア・アーカイブ生成器によって両方のコード片の生成のために使用されたそれらの生成命令が第２のソフトウェア・アーカイブ生成器に供給される。

【０００６】第１のソフトウェア・コンポーネント併合ユニットは、第１の署名付きコード片を生成するための生成命令を使用する。ユーザ側でそれらの生成命令を使用して第２の署名付きコード片を生成するので、生成命令がソフトウェア取得エンティティへ、ソフトウェア・プロバイダによって、好ましくは第２のソフトウェア・アーカイブ・ジェネレータと共に提供される場合に有利である。それにより、ユーザが、第２の署名付きコード片の正しい生成を許可する１組のツールを所有しているという、よりよい保証が存在する。

【０００７】ソフトウェア・プロバイダ側のシステムがさらに、秘密鍵へのアクセスを有する署名ユニットを含むことができる。コード片は、秘密鍵を使用して署名される。公開／秘密鍵システムは、非常に広く行き渡り、よく知られている暗号化システムであり、したがって実施しやすく使用しやすい。

【０００８】差分コードが、好ましくは第１のソフトウェア・アーカイブ生成器によって作成され、第１のソフトウェア・アーカイブ生成器が第２の署名付きコード片を生成する場合、有利である。差分コードが、第２のコード片に達する方法のステップをリフェルセット（refelcets）するので、差分ファイルに入れられる情報が自動的に作成され、生成処理中に使用可能である。

【０００９】既存の第１の署名付きコード片から第２の署名付きコード片に達するソフトウェア取得エンティティのための方法であって、それにより両方のコード片が、ソフトウェア・プロバイダ側で、第１のソフトウェア・アーカイブ生成器を使用することによって、生成命令の使用下で生成されており、この方法は、第１の署名付きコード片から第２の署名付きコード片に達するために必要なステップを含む差分コードの送達のために、コード修正要求をソフトウェア・プロバイダに送信するステップと、差分コードを受信し、第２のソフトウェア・アーカイブ生成器を使用することによって差分コードを第１の署名付きコード片に結合し、それにより第２の署名付きコード片を生成するステップとを含む。第１のソフトウェア・アーカイブ生成器によって両方のコード片の生成のために使用されたそれらの生成命令が第２のソフトウェア・アーカイブ生成器に供給される。ユーザには、完全に新しい第２の署名付きコードではなく、小さい差分コードのみを使用する利点がある。差分コードを、ネットワークを介してダウンロードすることで、著しく少ない時間しかかからない可能性がある。ソフトウェア・プロバイダにとっても、このことが当てはまる。差分ファイルを提供するためのコストも、完全バージョンの第２の署名付きコード片のコストよりも低くなるであろう。

【００１０】提案した方法は、提案した方法を実行するためのプログラム・コードを含むコンピュータ・プログラム製品の形式で実現し、市場に出すことができる。このコンピュータ・プログラム製品を、コンピュータ可読媒体上に格納することができる。

【００１１】本発明を理解するため、まず、既存の状況についてより詳細に説明する。普通または正当な人物など、あるエンティティが特定のコード片を取得しており、この人物を以下でユーザとも呼び、この人物がこのコードをコード片の異なる片で置き換えることを望むと仮定し、両方のコードをソフトウェアとも呼び、先に取得されたソフトウェア片の更新されたバージョンであることが好ましい。以下では、具体的な例を挙げるために、ユーザが、自分がすでに取得したソフトウェアの更新を有することを望むと仮定する。典型的には、ユーザが、すでに取得したソフトウェア片の作成元に更新を要求する。次いで、ソフトウェアの作成者が、更新された完全バージョンのソフトウェア、あるいは、まだ更新されていないソフトウェアの存在において実行されているときに、次いで既存のソフトウェアを更新されたソフトウェアに変更するソフトウェアを、ユーザに送信することができる。署名付きソフトウェアの場合、すなわち、ソフトウェアの完全性およびオリジナリティを、秘密／公開鍵暗号化方式を使用して検証することができる場合、このような更新では今まで、ソフトウェアの作成者が完全バージョンの更新を送信する必要があり、これは、先に取得されたソフトウェアを更新と共に組立てることにより、実際にすべての場合において、検証ツールによってオリジナルでないと認識されるであろう異なる署名に通じるからである。しかし、これは、ユーザが署名付きソフトウェアの署名を使用して、オリジナル、すなわち、無許可の手によって触れられていないソフトウェアを、ある無許可のエンティティによって修正されたソフトウェアから区別できるようにするためである。ユーザはこの署名の利点を、更新が行われるときにも持続することを望む。ユーザはクライアントとも呼ばれ、ソフトウェア・プロバイダはサーバとも呼ばれる。これは、典型的な実現では、ソフトウェア・プロバイダが自分で行うのではなく、クライアント・システムを動作するユーザに所望の更新コードを提供するために必要なステップを自動的に実行するサーバなど、コンピュータ・システムにおいて自動化された提供機能を有する可能性があるからである。ユーザ側では、更新要求に加えて最後の更新の実行を、クライアントと呼ばれる、コンピュータ化されたシステムにおいて自動化することができる。

【００１２】更新されるコードのための暗号化署名の再作成を許可する、暗号により保護された差分ファイルを使用する解法を提案する。クライアント側ソフトウェア

の署名の再作成のために提示した手法は、いかなる新しい暗号化機能性をもクライアント側で使用する必要がないが、これをサーバ側、およびサードパーティ実行環境、たとえば、ウェブブラウザに制限するという利点を有する。この一般の戦略において構築されたソフトウェアは、暗号化の方法を使用しないので、いかなるエクスポート制限も受けないという利点を有する。さらに、これは明らかにより安全であり、これは、署名鍵、すなわち、ソフトウェア・プロバイダの秘密鍵が、クライアント側で署名を再作成するために必要とされないからである。

【００１３】本発明は、インターネットなど本質的に不確かな媒体上の分野において、すでにインストールされ、配備されている署名付きアプリケーションを、安全に更新するための新しい手法を使用する。これは、上記の目的をとても効率的な方法で達成可能にする様々な技術を包含する。本質において、この概念について、オープン・セキュリティ・プロトコル、および、ＪＡＲまたはＣＡＢなど署名付き文書フォーマットに適用された識別技術と結合して、オンライン・バージョニング検査を使用するために説明することもできる。

【００１４】

【発明の実施の形態】以下で、本発明の様々な例示的実施形態について説明する。

【００１５】第１の実施形態

図１には、いくつかの署名付きコード片１１、１２、１３を含むデータベースを含む記憶装置１が示されている。これらの署名付きコード片１１、１２、１３は、第１のソフトウェア・アーカイブ生成器２によって生成されており、これもＡＧで示す。第１のソフトウェア・アーカイブ生成器２は、第１のソフトウェア・コンポーネント併合ユニット２１を含み、これが生成命令８を使用する。

【００１６】これはさらに署名ユニット２２を含み、これが秘密鍵１４へのアクセスを有する。図示の例では、４つのソフトウェア・コンポーネント９をＡ、Ｂ、Ｃ、Ｄで示し、これらが第１のソフトウェア・アーカイブ生成器２へ送り込まれ、これが生成命令８およびバージョン命令２６に従って、これらのソフトウェア・コンポーネント９をアセンブリに組み立て、これをアーカイブまたはコンテナとも呼ぶ。

【００１７】各コード片１１、１２、１３について、これらのバージョン命令２６が、何がその特定のコード片１１、１２、１３に含まれるか、および、それがどのようにどこで含まれるかについて、コード特有の情報を含む。よって、ここでは、第１の署名付きコード片１１について、これは規則ＲＶ１を含み、これは３つのコンポーネントＡ、Ｂ、Ｃが厳密にこの順序で入れられることを示す。第２の署名付きコード片１２について、これは規則ＲＶ２を含み、これは２つのコンポーネントＡ、Ｃが厳密にこの順序で入れられることを示す。第３の署名付きコード片１３について、これは規則ＲＶ３を含み、これは３つのコンポーネントＡ、Ｄ、Ｃが厳密にこの順序で入れられることを示す。生成命令８はより汎用の種類であり、いずれの署名付きコード片１１、１２、１３の生成にも適切である命令を与える。次いで、このアセンブリが、秘密鍵１４を使用して、署名ユニット２２によって署名される。この生成は、ここでは３つの異なるバージョンに通じる。すなわち、第１のバージョンＶ１．０、第２のバージョンＶ２．０および第３のバージョンＶ３．０である。第１のバージョンＶ１．０は署名Ｓｉｇ１１を有し、これにより第１の署名付きコード片１１を構築する。第２のバージョンＶ２．０は署名Ｓｉｇ１２を有し、これにより第２の署名付きコード片１２を構築する。第３のバージョンＶ３．０は署名Ｓｉｇ１３を有し、これにより第３の署名付きコード片１３を構築する。この具体的な例では、第１のバージョンＶ１．０が３つのソフトウェア・コンポーネントＡ、Ｂ、Ｃおよび署名Ｓｉｇ１１からなり、第２のバージョンＶ２．０が２つのソフトウェア・コンポーネントＡ、Ｃおよび署名Ｓｉｇ１２のみからなり、第３のバージョンＶ３．０が３つのソフトウェア・コンポーネントＡ、Ｄ、Ｃおよび署名Ｓｉｇ１３からなる。

【００１８】上記で説明したように、これらの３つのコード片１１、１２、１３が記憶装置１に格納される。記憶装置１が差分コード生成器１０に接続され、これがその出力を出力ユニット３へ送達し、これが入力ユニット２４に結合される。ここまでは、ソフトウェア・プロバイダ２５側のインフラストラクチャについて説明した。

【００１９】ソフトウェア取得エンティティ２０は、ユーザ２０とも呼ばれ、この側に第２のソフトウェア・アーカイブ生成器７が存在し、これはソフトウェア・コンポーネント・セパレータ２３および第２のソフトウェア・コンポーネント併合ユニット２７を含み、これが、ソフトウェア・プロバイダ側のものと等しい生成命令８の使用下で動作可能である。ユーザ２０は第１の署名付きコード片１１を有し、これを第２の署名付きコード片１２に修正することを望む。したがって、ユーザ２０が、どの更新が望まれるかについての情報を与える識別子が添付されるコード修正要求１６を、ソフトウェア・プロバイダ２５へ、入出力ユニット６を介して入力ユニット２４に送信する。ここでは、識別子が、ユーザ２０がソフトウェアＳＷ１のバージョンＶ１．０からバージョンＶ２．０への更新を必要とすることを示す。識別子は、現在の機能セットを記述するバージョン番号にすることができ、いかなる文字列、または数値にすることもできる。

【００２０】コード修正要求１６が、ソフトウェア・プロバイダ２５の入力ユニット２４において受信され、その一式において、差分コード生成器１０が２つのバージ

ョンＶ１．０およびＶ２．０を比較し、対応する差分コードＤＶ（ＳＷ１）を生成し、これは本明細書では第１の差分コード４、すなわちＤＶ１２（ＳＷ１）である。このコードの内容は、第１のバージョンＶ１．０を第２のバージョンＶ２．０へ修正するための命令であり、ソフトウェア・コンポーネントＢが除去され、署名Ｓｉｇ１１が除去されて署名Ｓｉｇ１２で置き換えられることを示す。よって、差分コード生成器は、内容における差分、および２つの署名付きコンテナ・ファイル・ツール、すなわち第１の署名付きコード片１１および第２の署名付きコード片１２の署名を抽出するためのツールである。更新された署名付きのコンテナに含まれたすべてのエントリの厳密な順序が記録される。

【００２１】よって、第１の差分コード４は、第１のコード片１１から開始して第２のコード片１２に達するために必要なすべてのステップを含む。これらのステップは、ユーザ側のソフトウェア・アーカイブ生成器７が、第２の署名付きコード片１２を生成するために必要である精度と共に与えられ、これは、最後には、第２のコード片１２に属する署名Ｓｉｇ１２が、ユーザ側で更新されたコード片のための正しい署名であるという方法による。差分ファイル４がユーザ２０のためのバージョン命令２６を反映して、ユーザ２０側で生成命令８のみが必要とされるようにする、ということができる。

【００２２】特定のコード片の署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３の検証が、コードの内部構造に依存する特定の結果を出し、これは典型的にはハッシュ技術が署名に使用されるからである。署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３があるコード片について正しく識別され、検証されるために、よってこのコード片の内部構造が、署名されたコード片の内部構造と等しくなければならない。これは、秘密／公開鍵暗号システムが、特別に高い確率により、２つの異なるコード片についての結果が異なることを保証するからである。これは、典型的に使用される、固有の非対称に基づく暗号化方式のいずれにも当てはまり、暗号化されたファイルを生成するために使用される時間が、典型的には数桁分、暗号化されたファイルから暗号化されていないファイルに達するために要するであろう時間よりもはるかに短い。後者の時間は、実際の場合に人間の寿命を越え、よってこれが暗号化の期間において安全として定義される。

【００２３】よって、ユーザ側で第１のソフトウェア片１１を第２のソフトウェア片１２に修正するための処理が、結果の第２の署名付きコード片１２が、元の第２の署名付きコード片１２がソフトウェア・プロバイダ側で生成され署名されたときに有したものと厳密に同じ内部構造を有することを保証する方法で行われることが、保証されなければならない。第１の差分コード４が生成命令８およびコンポーネント併合ユニット２１と共にツールボックスを形成し、これが、ユーザ側の生成処理がソフトウェア・プロバイダ側の生成処理と合致することを保証する。この合致により、ユーザ側で生成されるときの第２の署名付きコード片１２の内部構造が、ソフトウェア・プロバイダ側で格納されるときの第２の署名付きコード片１２の内部構造と等しいことを保証する。

【００２４】第１の差分コード４が入出力ユニット６で受信された後、これが第２のソフトウェア・アーカイブ生成器７へ転送される。ここで、第１のコード片１１が最初にそのソフトウェア・コンポーネント９に分離され、これらが次いで、生成命令８および第１の差分コード４に含まれた命令に従って、コンポーネント併合ユニット２１によって併合される。この結果は、その署名Ｓｉｇ１２を有する第２の署名付きコード片１２である。よって、公開鍵１５の使用下での暗号化検証ユニットによる後続の検証は、肯定的な検証結果１９の結果にならなければならない。

【００２５】この方法では、第２のコード片１２全体ではなく、差分コード４のみを伝送すればよい。大きいコード片の場合、第２のコード片１２と差分コード４の間のサイズにおける差は、たとえば電子伝送用の転送コストを考えると、非常に大きく莫大な影響となる可能性がある。インターネットのような電子ネットワークを介した小さいコード片の伝送は、大きいサイズのコード片の伝送と比較して、要する時間がより少なく、より安価で、割り込みおよびデータ損失の傾向が少ない。この伝送は、電子ネットワークを介して行う必要はなく、印刷された形式、プリンタおよびスキャナ、メールシステムまたは手による搬送において、あるいは光ネットワークを介して、あるいは磁気的に格納されたデータとしてなど、いかなる人または機械可読形式において、いかなる適切な媒体を介して行うこともできる。差分コード４は、更新または差分ファイルとも呼ばれる。

【００２６】第２のソフトウェア・アーカイブ生成器７は、更新４を処理することができ、すなわち、新しい更新４を、すでにユーザ２０側にインストールされた既存の第１の署名付きコード片１１に併合する。この利点は、更新４が、クライアント側の第２のソフトウェア・アーカイブ生成器７によって、更新中のソフトウェア１１に署名するときにサーバ側に存在するものと厳密に同じコード・コンテナ構造を再作成するために厳守される、厳密な順序の内容を含むという事実にある。これにより、署名および内容に関する限り、クライアント側で、更新されたソフトウェア１２をそのままで正しく再作成することができる。

【００２７】第２の実施形態

上記で説明した例を拡張して、ユーザ２０が、第１の署名付きコード片１１から第３の署名付きコード片１３に達することを望むと仮定する。この更新では、原則として２つのステップが必要であり、すなわち、第１の署名付きコード片１１から第２の署名付きコード片１２に達

するための、第１の差分コード４を使用する第１の更新、および、第２の署名付きコード片１２から第３の署名付きコード片１３に達するための、第２の差分コード５を使用する第２の更新である。このためのいくつかの可能性が存在する。

【００２８】ａ）差分コードが、厳密に、第１の署名付きコード片１１から第３の署名付きコード片１３への移行のために決定される。差分コード生成器１０は、これらの２つの署名付きコード片１１、１３を比較することによって、これを行うことができる。

【００２９】ｂ）ユーザ２０が２つの差分コード４、５を受信し、一方の更新が、第２のソフトウェア・アーカイブ生成器７において他方の更新の後に実行される。しかし、これはあまり的確でないと見なされ、ユーザ２０に負担を与え、ユーザが、より不体裁でおそらくより遅い２つの更新に対処しなければならない。第１の更新における動作は、後の更新がこれらを取り消すので冗長的になっており、差分コード４、５がネットワークを介して伝送された場合に時間および帯域幅が無駄になる。この状況は、全体のアップグレードを実行するために必要な中間更新の数が増加するにつれて悪化する。

【００３０】ｃ）２つの差分コード４、５が全体の差分コードに結合され、次いでこれがユーザ２０に伝送される。

【００３１】差分コード４、５は、ユーザ２０の要求時に決定される必要はないが、その前のいかなる時点で所定にすることもでき、たとえばデータベースにも格納することができる。コード修正要求への応答を、もちろん、差分コード４、５が事前生成されて以前に格納されたときよりも速くすることができる。しかし、非常に多数のｎ個の異なるバージョンＶｘ．ｘの場合、この方法では、あらゆる可能な更新の組み合わせが事前生成されて格納されている場合、非常に多数の更新差分コード４、５、すなわち理論上はｎ＊（ｎ－１）／２個もの異なる差分コード４、５となる。簡素化されたスキームは、差分コード４を可能な組み合わせの部分集合から事前生成するのみとなる。一連の後続の更新の場合、この集合は、各バージョンＶｘ．０からその後続のバージョンＶ（ｘ＋１）．０までの差分コード４、５を含むことができる。これにより、差分コード４、５の連鎖が事前生成され、これがｎ－１個の異なる差分コード４、５のみを含む。いくつかの更新のステップを包含する更新の要求、すなわち差分コード４、５が到着する場合、対応する差分コード４、５を、上記のｃ）で示したように結合することができる。現実的な場合、このような更新要求の確率は、典型的には、１つのバージョンからその後続のバージョンへの更新要求の確率よりもはるかに低い。

【００３２】ユーザ２０とソフトウェア・プロバイダ２５の間の通信を、認証スキームを使用して行うことができる。２つの関係者、すなわち、ユーザ２０およびソフトウェア・プロバイダ２５は、それにより、相手方が信用できるかどうか、真の相手方であるように装っている別の相手方ではないかどうかを決定することができる。この認証処理が通信を容易にし、更新の後に検証で不正な更新の試みを認識する結果となる状況を回避する。

【００３３】認証スキームは、以下のように統合することができる。

【００３４】ソフトウェア・プロバイダ側のサーバ２５が、クライアント認証を必要とするＳＳＬプロトコルなど、認証プロトコルを動作中である。

【００３５】サーバ２５が、署名付きコード片１１、１２、１３のリポジトリを維持し、これはソフトウェア・ファイルとも呼ばれ、先にユーザ２０、すなわち顧客に渡されたすべての可能なバージョンを記述するバージョン番号によって索引付けされる。サーバ２５は、所与のソフトウェア・ファイルを別のソフトウェア・ファイル、好ましくは最新のソフトウェア・バージョンに変換するために必要な、すべてのデータおよびすべての署名関連情報を含む署名付き差分ファイル４も、格納しておくことができる。

【００３６】クライアント側で、ＳＳＬクライアント・コードなど、クライアント・コードを動作し、これを使用して上記サーバ２５への安全な接続を確立することができる。

【００３７】クライアント側の認証ソフトウェアがさらに、元のソフトウェア配布に含まれた暗号化証明書および鍵を、たとえば、元のソフトウェア・ファイル・コードを搬送する製品ＣＤに付属している個別化されたディスク上に含み、これを使用して、ソフトウェア配布サーバ２５によりクライアント認証プロトコルを実行することができる。このように、クライアント２０は、通信するサーバ２５が本物のサーバ２５であると確かめることができ、サーバ２５には、通信するクライアント２０にソフトウェア更新４の資格があることが分かる。

【００３８】ソフトウェア更新４が、この安全な接続を介してクライアント２０へ伝送される。

【００３９】クライアント２０が、更新の内容を検査する。すなわち、詳細には、ソフトウェア更新４、５に含まれたバージョン情報が、それが適切であるか、場合によってはクライアント２０上に存在する現在のバージョンに等しいかどうかについて検査される。

【００４０】次いで、クライアント側のソフトウェアが必要なすべてのデータを抽出し、これを、すでにクライアント２０に存在する既存のソフトウェア１１に結合する。詳細には、すべての署名付きコード・コンテナの内容の順序が適切にリストアされ、署名が、サーバ側で最初に作成されたそれらの機能性および正当性を達成するようにする。

【００４１】最後のステップで、サーバ２５から受信さ

れた更新ファイル４、５に含まれたすべての署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３が、適切なファイル、すなわち、たとえばＮｅｔｓｃａｐｅ　ＪＡＲまたはＭｉｃｒｏｓｏｆｔ　ＣＡＢファイルなど、署名付きコード・コンテナに適用される。

【００４２】上で概説したこの手法を、継続された更新を必要とし、たとえばＭｉｃｒｏｓｏｆｔ　ＣＡＢまたはＮｅｔｓｃａｐｅ　ＪＡＲなど、上述のファイル・フォーマットの１つにより署名される、いかなる形式のデータにも適用することができる。

【００４３】上記で説明したシナリオを、非コネクション型シナリオで増すか、あるいは置き換えることができ、この場合は差分ファイル自体が暗号化により保護される。これは、ソフトウェア配布サーバ２５へのオンライン接続が確立できないか、あるいはそれが望ましくない場合、意味をなす。この場合、サーバ側のソフトウェア配布機能が、暗号化署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３を差分ファイル４、５に適用し、これがクライアント側の更新ソフトウェアによって、上で概説したオンラインＳＳＬ接続確立におけるセキュア・セッション確立に相当する方法において検査される。この手法に適切なファイル・フォーマットは、内部で処理することができるものと同じフォーマットであり、すなわち、たとえば、Ｎｅｔｓｃａｐｅ／ＳｕｎのＪＡＲまたはＭｉｃｒｏｓｏｆｔ　ＣＡＢである。

【００４４】認証がなくとも、署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３が、ある程度のセキュリティを提供し、これは更新処理によって損なわれない。検証はすなわち、いかなる理由であれ、更新された第２のコード片１２の一部である署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３が正しくなかった場合、すなわち、公開鍵１５を使用することによって計算される、予想された署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３と合致しなかった場合、否定的な結果を出す。このような場合、合致しなかった署名Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３が、何かがうまく行かなかったこと、および、更新が信用されないことを信号で通知する。よって、ユーザ２０が、更新された第２のコード片１２を使用しないように決定することができ、これは、ある無許可の修正がセキュリティ上の問題を導入した可能性があり、これがユーザ２０を害する可能性があるからである。ユーザ２０が第１の署名付きコード片１１のバックアップ・コピーを保持していた場合、更新されたバージョン１２を削除することができ、なお最初の第１の署名付きコード片１１を使用することができる。

【００４５】説明した実施形態は、部分的に、ならびに全部を組み合わせることができる。理解のために、更新に言及するとき、本発明はソフトウェアの更新に制限されるものではなく、第１の署名付きコード片１１から第２、第３のコード片１２、１３などへのステップを実行するように適用することができるが、第１の署名付きコード片１１のために作成された署名の利点および正当性を維持することに留意されたい。ソフトウェア・アーカイブ生成器２、７として、いかなるコード片のそれぞれのそのハードウェア・バージョンは、それが、それらを単一のソフトウェア製品またはコンピュータ・プログラムとして単一化するために、詳細には、それを顧客または他のエンティティに送達するために、ソフトウェア・コンポーネントを一緒にまとめる機能を実行することを意味する。

【００４６】本発明を、ハードウェア、ソフトウェア、またはこれらの組み合わせにおいて実現できることは、当業者には明らかである。これは、集中方式においてある単一のコンピュータ・システム上で、あるいは、分散方式において実施することもでき、この場合は、異なる要素がいくつかの相互接続されたコンピュータまたはコンピュータ・システムに渡って広がっており、それにより、いかなる種類のコンピュータ・システムも、あるいは本明細書で説明した方法を実行するように適合された他の装置が適する。典型的なハードウェアおよびソフトウェアの組み合わせを、ロードかつ実行中に、本明細書で説明した方法を実行するようにコンピュータ・システムを制御する、コンピュータ・プログラムを有する汎用コンピュータ・システムにすることができる。本発明をコンピュータ・プログラム製品に埋め込むこともでき、これは、本明細書で説明した方法の実施を可能にするすべての機能を含み、コンピュータ・システムにロードされたときにこれらの方法を実行することができる。

【００４７】この文脈におけるコンピュータ・プログラム手段またはコンピュータ・プログラムは、情報処理機能を有するシステムに、特定の機能を、直接あるいは以下のａ）別の言語、コードまたは表記法への変換、ｂ）異なる材料形式における複製のいずれかあるいは両方の後に実行させるように意図された命令セットの、いかなる言語、コードまたは表記法における、いかなる表現をも意味する。

【００４８】まとめとして、本発明の構成に関して以下の事項を開示する。

【００４９】（１）ソフトウェア取得エンティティ（２０）が既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達することができるようにする、ソフトウェア・プロバイダ（２５）のための方法であって、両方のコード片（１１、１２、１３）は、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、前記第１の署名付きコード片（１１）から前記第２の署名付きコード片（１２、１３）に達するために必要なステップを含む差分コード（４、５）を前記ソフトウェア取得エンティティ（２０）に提供するステップを含み、この差分コード（４、５）は、前記ソフ

トウェア取得エンティティ（２０）側で、第２のソフトウェア・アーカイブ生成器（７）によって前記第１の署名付きコード片（１１）と結合されて前記第２の署名付きコード片（１２、１３）を生成するように使用可能であり、それにより、前記第１のソフトウェア・アーカイブ生成器（２）によって両方のコード片（１１、１２、１３）の前記生成のために使用されたそれらの生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給される方法。

（２）前記生成命令（８）が、前記ソフトウェア・プロバイダ（２５）によって、好ましくは前記第２のソフトウェア・アーカイブ生成器（７）と共に前記ソフトウェア取得エンティティ（２０）に提供される、上記（１）に記載の方法。

（３）前記コード片（１１、１２、１３）が、秘密鍵（１４）を使用して署名される、上記（１）または（２）に記載の方法。

（４）前記署名付きコード片（１１、１２、１３）が、前記ソフトウェア・プロバイダ（２５）側の記憶装置（１）に格納される、上記（１）ないし（３）のいずれか一項に記載の方法。

（５）前記差分コード（４、５）が、好ましくは前記第１のソフトウェア・アーカイブ生成器（２）によって作成され、前記第１のソフトウェア・アーカイブ生成器（２）が前記第２の署名付きコード片（１２、１３）を生成する、上記（１）ないし（４）のいずれか一項に記載の方法。

（６）３つ以上のコード片（１１、１２、１３）が格納される場合、前記差分コード（４、５）が前記コード片（１１、１２、１３）の部分集合の間でのみ生成される、上記（１）ないし（５）のいずれか一項に記載の方法。

（７）前記第１のコード片（１１）から前記第２のコード片（１３）に達するために、いくつかの差分コード（４、５）が必要とされ、これらの差分コード（４、５）が単一の差分コードに併合されて前記ソフトウェア取得エンティティ（２０）に提供される、上記（６）に記載の方法。

（８）前記第１および第２のコード片（１１、１２、１３）が、前記ソフトウェア・プロバイダ（２５）側で、前記ソフトウェア取得エンティティ（２０）から受信された要求（１６）から対応する識別子を導出することによって識別される、上記（１）ないし（７）のいずれか一項に記載の方法。

（９）既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達するソフトウェア取得エンティティ（２０）のための方法であって、両方のコード片（１１、１２、１３）は、ソフトウェア・プロバイダ（２５）側で、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令

（８）の使用下で生成されており、前記第１の署名付きコード片（１１）から前記第２の署名付きコード片（１２、１３）に達するために必要なステップを含む差分コード（４、５）の送達のために、コード修正要求（１６）を前記ソフトウェア・プロバイダ（２５）に送信するステップと、前記差分コード（４、５）を受信するステップと、第２のソフトウェア・アーカイブ生成器（７）を使用することによって前記差分コード（４、５）を前記第１の署名付きコード片（１１）と結合し、それにより前記第２の署名付きコード片（１２、１３）を生成するステップとを含み、それにより、前記第１のソフトウェア・アーカイブ生成器（２）によって両方のコード片（１１、１２、１３）の前記生成のために使用されたそれらの生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給される方法。

（１０）前記生成命令（８）が前記ソフトウェア・プロバイダ（２５）から、好ましくは前記第２のソフトウェア・アーカイブ生成器（７）と共に受信される、上記（９）に記載の方法。

（１１）前記コード片（１１、１２、１３）が、秘密鍵（１４）を使用することによって署名され、署名（Ｓｉｇ１１、Ｓｉｇ１２、Ｓｉｇ１３）が、対応する公開鍵（１５）を使用することによって検証可能である、上記（９）または（１０）に記載の方法。

（１２）前記第１および第２のコード片（１１、１２、１３）が、前記コード修正要求（１６）において対応する識別子を与えることによって前記ソフトウェア取得エンティティ（２０）によって識別される、上記（９）ないし（１１）のいずれか一項に記載の方法。

（１３）上記（１）ないし（１２）のいずれか一項に記載の方法を実行するためのプログラム・コード手段を含む、コンピュータ・プログラム製品。

（１４）コンピュータ可読媒体上に格納された前記プログラム・コード手段を含む、上記（１３）に記載のコンピュータ・プログラム製品。

（１５）ソフトウェア取得エンティティ（２０）が、既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達することができるようにするコード修正イネーブラであって、両方のコード片（１１、１２、１３）は、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、前記第１の署名付きコード片（１１）から前記第２の署名付きコード片（１２、１３）に達するために必要なステップを含む差分コード（４、５）を生成するための差分コード生成器（１０）を含み、この差分コード（４、５）は、前記ソフトウェア取得エンティティ（２０）側で、第２のソフトウェア・アーカイブ生成器（７）によって前記第１の署名付きコード片（１１）と結合されて前記第２の署名付きコード片（１２、１３）を生成するように使用可能であ

り、それにより、前記生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給され、さらに、前記ソフトウェア取得エンティティ（２０）に前記差分コード（４、５）を提供するための出力ユニット（３）を含む、コード修正イネーブラ。

（１６）前記差分コード（４、５）の送達のためのコード修正要求（１６）を前記ソフトウェア取得エンティティ（２０）から受信するための入力ユニット（２４）をさらに含む、上記（１５）に記載のコード修正イネーブラ。

（１７）前記コード片（１１、１２、１３）を生成命令（８）の使用下で生成するための第１のソフトウェア・アーカイブ生成器（２）をさらに含む、上記（１５）または（１６）に記載のコード修正イネーブラ。

（１８）既存の第１の署名付きコード片（１１）から第２の署名付きコード片（１２、１３）に達するためのコード修正デバイスであって、両方のコード片（１１、１２、１３）は、ソフトウェア・プロバイダ（２５）側で、第１のソフトウェア・アーカイブ生成器（２）を使用することによって、生成命令（８）の使用下で生成されており、受信された差分コード（４、５）を前記第１の署名付きコード片（１１）と結合し、それにより前記第２の署名付きコード片（１２、１３）を生成するための第２のソフトウェア・アーカイブ生成器（７）を含み、それにより、前記第１のソフトウェア・アーカイブ生成器（２）によって両方のコード片（１１、１２、１３）の前記生成のために使用されたそれらの生成命令（８）が前記第２のソフトウェア・アーカイブ生成器（７）に供給される、コード修正デバイス。

（１９）コード修正要求（１６）を前記ソフトウェア・プロバイダ（２５）に送信し、前記差分コード（４、５）を受信するための入出力ユニット（６）をさらに含む、上記（１８）に記載のコード修正デバイス。

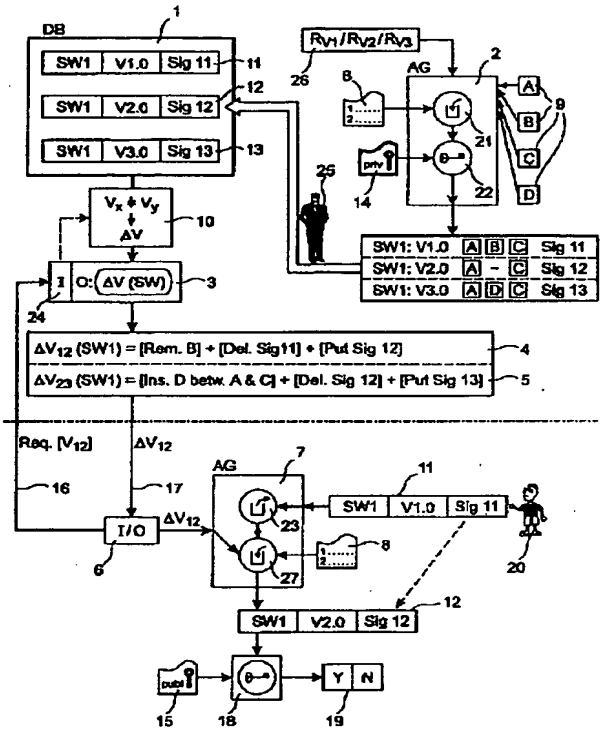
【図面の簡単な説明】

【図１】ユーザとソフトウェア・プロバイダの間のコードの交換において関係するステップおよびユニットの概略図である。

【符号の説明】

１　記憶装置
２　第１のソフトウェア・アーカイブ生成器
３　出力ユニット
４　差分コード
５　差分コード
７　第２のソフトウェア・アーカイブ生成器
８　生成命令
１０　差分コード生成器
１１　第１の署名付きコード片
１２　第２の署名付きコード片
１３　第２の署名付きコード片
１４　秘密鍵
１５　公開鍵
１６　コード修正要求
２０　ソフトウェア取得エンティティ
２４　入力ユニット
２５　ソフトウェア・プロバイダ
Ｓｉｇ１１　署名
Ｓｉｇ１２　署名
Ｓｉｇ１３　署名

【図１】



フロントページの続き

(51) Int.Cl.[7]		識別記号	FI		テーマコード(参考)
G06F	13/00	530	G09C	1/00	640B
	15/00	330	G06F	9/06	660G
G09C	1/00	640			640A

(72)発明者　ミヒャエル・ベンチェ
スイス　シー・エイチ8135　ラングナウ　アムアルビス　シルワルト・ストラーセ　4
(72)発明者　ペーター・ブラー
スイス　シー・エイチ8803　リュシュリコン　ミュエレストラーセ　39
(72)発明者　トーマス・エイリッヒ
スイス　シー・エイチ8804　アウ　ゾップフストラーセ　16
(72)発明者　フランク・ヘリング
スイス　シー・エイチ8006　チューリッヒ　クルマンストラーセ　39
(72)発明者　トーマス・ヴァイゴルト
スイス　シー・エイチ8134　アドリスヴィル　ブッテナウストラーセ　20

Fターム(参考)　5B017 AA03 BA07 CA15
5B076 AB10 BB06 FA00
5B082 GA04 GA11
5B085 AE13 BG07
5J104 AA09 LA03 NA02